

FFFFFFF	000000000	RRRRRRRRRRR	RRRRRRRRRRR	TTTTTTTTTTTTT	LLL
FFFFF	000000000	RRRRRRRRRRR	RRRRRRRRRRR	TTTTTTTTTTTTT	LLL
FFFFF	000000000	RRRRRRRRRRR	RRRRRRRRRRR	TTTTTTTTTTTTT	LLL
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000	000	RRR	RRR	TTT
FFF	000000000	RRR	RRR	RRR	LLLLLLLLLLLL
FFF	000000000	RRR	RRR	RRR	LLLLLLLLLLLL
FFF	000000000	RRR	RRR	RRR	LLLLLLLLLLLL

FILEID**FORDELETE

J 15

FFFFFFFFF 000000 RRRRRRRR DDDDDDDD EEEEEEEEEE LL EEEEEEEEEE TTTTTTTTTT EEEEEEEEEE
FFFFFFFFF 000000 RRRRRRRR DDDDDDDD EEEEEEEEEE LL EEEEEEEEEE TTTTTTTTTT EEEEEEEEEE
FF 00 00 RR RR DD DD EE LL EE TT EE
FF 00 00 RR RR DD DD EE LL EE TT EE
FF 00 00 RR RR DD DD EE LL EE TT EE
FF 00 00 RR RR DD DD EE LL EE TT EE
FFFFFFFFF 00 00 RRRRRRRR DD DD EEEEEEEEEE LL EEEEEEEEEE TT EEEEEEEEEE
FFFFFFFFF 00 00 RRRRRRRR DD DD EEEEEEEEEE LL EEEEEEEEEE TT EEEEEEEEEE
FF 00 00 RR RR DD DD EE LL EE TT EE
FF 00 00 RR RR DD DD EE LL EE TT EE
FF 00 00 RR RR DD DD EE LL EE TT EE
FF 00 00 RR RR DD DD EE LL EE TT EE
FF 00 00 RR RR DDDDDDDD EEEEEEEEEE LLLLLLLLLL EEEEEEEEEE TT EEEEEEEEEE
FF 000000 RR RR DDDDDDDD EEEEEEEEEE LLLLLLLLLL EEEEEEEEEE TT EEEEEEEEEE
FF 000000 RR RR DDDDDDDD EEEEEEEEEE LLLLLLLLLL EEEEEEEEEE TT EEEEEEEEEE
LL IIIII SSSSSSSS
LL IIIII SSSSSSSS
LL II SS SSSSSS
LL LLLL LLLL IIIII SSSSSSSS
LL LLLL LLLL IIIII SSSSSSSS

```
1 0001 0 MODULE FOR$DELETE ( ! DELETE statement processor
2 0002 0 IDENT = '1-002' ! Edit: SBL1002
3 0003 0 )
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 ****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 ****
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1 * FACILITY: FORTRAN Language Support Library
33 0033 1
34 0034 1 * ABSTRACT:
35 0035 1
36 0036 1 * Contains routines to implement FORTRAN DELETE for relative
37 0037 1 * and indexed organization files.
38 0038 1
39 0039 1 * ENVIRONMENT: User mode, AST reentrant
40 0040 1
41 0041 1 * AUTHOR: Steven B. Lionel, CREATION DATE: 14-May-1979
42 0042 1
43 0043 1 * EDIT HISTORY:
44 0044 1
45 0045 1 * 1-001 - Original. SBL 14-May-1979
46 0046 1 * 1-002 - Move ACTUALCOUNT declaration inside routine. SBL 15-June-1982
47 0047 1 --
```

```

49
50 0048 1 ! SWITCHES:
51 0049 1 ! :
52 0050 1 ! :
53 0051 1 ! SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
54 0052 1 ! :
55 0053 1 ! :
56 0054 1 ! :
57 0055 1 ! LINKAGES:
58 0056 1 ! :
59 0057 1 ! :
60 0058 1 REQUIRE 'RTLIN:OTSLNK'; ! Define all linkages
61 0487 1 !
62 0488 1 ! :
63 0489 1 ! TABLE OF CONTENTS:
64 0490 1 ! :
65 0491 1 ! FORWARD ROUTINE
66 0492 1 FOR$DELETE : CALL_CCB, ! Sequential access delete entry
67 0493 1 FOR$DELETE_D : CAL_CCB; ! Direct access delete entry
68 0494 1 !
69 0495 1 !
70 0496 1 ! :
71 0497 1 ! INCLUDE FILES:
72 0498 1 ! :
73 0499 1 ! :
74 0500 1 LIBRARY 'RTLSTARLE'; ! STARLET library for macros and symbols
75 0501 1 REQUIRE 'RTLML:FORERR'; ! FORTRAN error numbers
76 0569 1 REQUIRE 'RTLML:OTSLUB'; ! Logical Unit Block Definitions
77 0709 1 REQUIRE 'RTLML:OTSISB'; ! Inter-statement Block Definitions
78 0877 1 REQUIRE 'RTLIN:RTLPSECT'; ! P-SECT declaration macros
79 0972 1 REQUIRE 'RTLML:FORPAR'; ! Inter-module parameters
80 0995 1 !
81 0996 1 !
82 0997 1 ! MACROS:
83 0998 1 !
84 0999 1 !
85 1000 1 !
86 1001 1 ! EQUATED SYMBOLS:
87 1002 1 !
88 1003 1 !
89 1004 1 !
90 1005 1 ! PSECT DEFINITIONS:
91 1006 1 !
92 1007 1 !
93 1008 1 ! DECLARE_PSECTS (FOR):
94 1009 1 !
95 1010 1 !
96 1011 1 ! OWN STORAGE:
97 1012 1 !
98 1013 1 !
99 1014 1 !
100 1015 1 ! EXTERNAL REFERENCES:
101 1016 1 !
102 1017 1 !
103 1018 1 ! EXTERNAL ROUTINE
104 1019 1 FOR$SIOSTAT_HND, ! IOSTAT error handler
105 1020 1 FOR$SIG_STOP : NOVALUE, ! Signal_stop routine
106 1021 1 FOR$SIG_NO_LUB : NOVALUE, ! Signal without LUB

```

FOR\$DELETE
1-002

M 15

16-Sep-1984 00:17:29
14-Sep-1984 12:31:48

VAX-11 Bliss-32 v4.0-742
[FORRTL.SRC]FORDELETE.B32;1

Page 3
(2)

: 106 1022 1
: 107 1023 1

FOR\$CB_PUSH : JSB CB PUSH NOVALUE,
FOR\$CB_POP : JSB CB POP NOVALUE;
 ! Push a CCB
 ! Pop a CCB

```
109      1024 1 GLOBAL ROUTINE FOR$DELETE (
110          1025 1     UNIT,
111          1026 1     ERR_EQL
112          1027 1 ) :CALL_CCB =
113          1028 1
114          1029 1 ++
115          1030 1     FUNCTIONAL DESCRIPTION:
116          1031 1
117          1032 1     FOR$DELETE deletes the current record of the indexed or relative organization file
118          1033 1     open on unit UNIT.
119          1034 1
120          1035 1     CALLING SEQUENCE:
121          1036 1
122          1037 1     iostat.wl.v = FOR$DELETE (unit.rl.v [, err_eql.rlu.v])
123          1038 1
124          1039 1     FORMAL PARAMETERS:
125          1040 1
126          1041 1     unit           - The unit number on which to delete the record.
127          1042 1           There must be a "current record" on this unit.
128          1043 1     err_eql        - If absent or zero, all errors are signalled.
129          1044 1           - If 1, errors return an IOSTAT error code value.
130          1045 1
131          1046 1     IMPLICIT INPUTS:
132          1047 1
133          1048 1
134          1049 1
135          1050 1     CCB
136          1051 1
137          1052 1     IMPLICIT OUTPUTS:
138          1053 1
139          1054 1
140          1055 1
141          1056 1     NONE
142          1057 1
143          1058 1     ROUTINE VALUE:
144          1059 1
145          1060 1     An IOSTAT value.
146          1061 1
147          1062 1
148          1063 1
149          1064 1
150          1065 1
151          1066 1
152          1067 2
153          1068 2
154          1069 2
155          1070 2     BEGIN
156          1071 2
157          1072 2
158          1073 2     GLOBAL REGISTER
159          1074 2     CCB = 11: REF BLOCK [, BYTE];
160          1075 2
161          1076 2
162          1077 2
163          1078 2
164          1079 2
165          1080 2     LOCAL
166
167          1073 2     L_UNWIND_ACTION : VOLATILE,
168          1074 2     L_ERR_EQC_PRES : VOLATILE,
169          1075 2     STATUS;
170
171          1076 2     BUILTIN
172          1077 2     ACTUALCOUNT;
173          1078 2
174          1079 2
175          1080 2     ENABLE
```

```
: 166      1081  2      FOR$SIOSTAT_HND (L_UNWIND_ACTION, L_ERR_EQL_PRES);  
: 167      1082  2  
: 168      1083  2      |+ Determine if ERR_EQL is present.  
: 169      1084  2      |-  
: 170      1085  2      IF ACTUALCOUNT () GTR 1  
: 171      1086  2      THEN  
: 172      1087  2      L_ERR_EQL_PRES = .ERR_EQL  
: 173      1088  2      ELSE  
: 174      1089  2      L_ERR_EQL_PRES = 0;  
: 175      1090  2  
: 176      1091  2      |+ Unwind action is NO-OP (no LUB yet)  
: 177      1092  2      |-  
: 178      1093  2      L_UNWIND_ACTION= FOR$K_UNWINDNOP;  
: 179      1094  2  
: 180      1095  2  
: 181      1096  2  
: 182      1097  2      |+ Get a LUB for this unit. On return, CCB points to the  
: 183      1098  2      current control block.  
: 184      1099  2      |-  
: 185      1100  2      FOR$CB_PUSH (.UNIT, LUB$K_LUN_MIN);  
: 186      1101  2  
: 187      1102  2      |+ Unwind action is now to POP the CCB.  
: 188      1103  2      |-  
: 189      1104  2      L_UNWIND_ACTION = FOR$K_UNWINDPOP;  
: 190      1105  2  
: 191      1106  2  
: 192      1107  2      |+ If file is not indexed or relative organization, or is direct access,  
: 193      1108  2      signal error FOR$_DELEERR.  
: 194      1109  2      |-  
: 195      1110  2  
: 196      1111  2  
: 197      1112  2      |+  
: 198      1113  2      |+ If file is not indexed or relative organization, or is direct access,  
: 199      1114  2      signal error FOR$_DELEERR.  
: 200      1115  2      |-  
: 201      1116  2  
: 202      1117  2      IF NOT .CCB [LUB$V_NOTSEQORG] OR .CCB [LUB$V_DIRECT]  
: 203      1118  2      THEN  
: 204      1119  2      FOR$$SIGNAL_STO (FOR$K_DELEERR);  
: 205      1120  2  
: 206      1121  2      |+  
: 207      1122  2      |+ If file is read-only, signal error FOR$_WRIREAFIL.  
: 208      1123  2      |-  
: 209      1124  2  
: 210      1125  2      |+  
: 211      1126  2      |+ If file is read-only, signal error FOR$_WRIREAFIL.  
: 212      1127  2      |-  
: 213      1128  2  
: 214      1129  2  
: 215      1130  2      |+ Try to delete the current record. If we get an error, signal it.  
: 216      1131  2      |-  
: 217      1132  2  
: 218      1133  3      IF NOT $DELETE (RAB=.CCB)  
: 219      1134  2      THEN  
: 220      1135  3      BEGIN  
: 221      1136  3      WHILE .CCB [RAB$L_STS] EQL RMSS_RSA DO  
: 222      1137  4      BEGIN
```

```

: 223      1138   4      SWAIT (RAB=.CCB);
: 224      1139   4      $DELETE (RAB=.CCB);
: 225      1140   3      END;
: 226      1141   3      IF NOT .CCB [RABSL_STS]
: 227      1142   3      THEN
: 228          FOR$$SIGNAL_STO (
: 229          SELECTONEU .CCB [RABSL_STS] OF
: 230          SET
: 231          [RMSS CUR, RMSS RNL] : FOR$K_NO_CURREC;
: 232          [OTHERWISE] : FOR$K_DELERR;
: 233          TES);
: 234      1149   2      END;
: 235      1150   2
: 236      1151   2      + Return I/O system to previous state
: 237      1152   2      -
: 238      1153   2
: 239      1154   2
: 240      1155   2      FOR$SCB_POP ();
: 241      1156   2
: 242      1157   2      RETURN 0;           ! Success IOSTAT value
: 243      1158   1      END;

```

```

.TITLE FOR$DELETE
.IDENT \1-002\

.EXTRN FOR$IOSTAT_HND
.EXTRN FOR$$SIGNAL_STO
.EXTRN FOR$$SIG_NO_LUB
.EXTRN FOR$SCB_PUSH, FOR$SCB_POP
.EXTRN SYSSDELETE, SYSSWAIT

.PSECT _FORSCODE,NOWRT, SHR, PIC,2

      081C 00000
      54 00000000G 00 9E 00002    ENTRY FOR$DELETE. Save R2,R3,R4,R11      1024
      53 00000000G 00 9E 00009    MOVAB SYSSDELETE, R4
      5E          04 C2 00010    MOVAB FOR$$SIGNAL_STO, R3
                      7E D4 00013    SUBL2 #4, SP
                      04 AE 00015    CLRL L_ERR_EQL_PRES
                      CF DE 00018    CLRL L_UNWIND_ACTION
                      6C 91 0001D    MOVAL 12$, (FP)
                      06 1B 00020    CMPB (AP), #1
                      6E 08 00022    BLEQU 1$
                      02 11 00026    MOVL ERR_EQL, L_ERR_EQL_PRES
                      6E D4 00028 1$:    BRB 2$
                      01 D0 0002A 2$:    CLRL L_ERR_EQL_PRES
                      50 D4 0002E    MOVL #T, L_UNWIND_ACTION
      04  AE          04 AC 00030    CLRL RO
                      52 00000000G 00 16 00034    MOVL UNIT, R2
                      04 AE 0003A    JSB FOR$SCB_PUSH
                      03 E1 0003D    CLRL L_UNWIND_ACTION
      05  A1  AB          04          04 E1 00042    BBC #3, -95(CC), 3$
      05  FC  AB          37 DD 00047 3$:    BBC #4, -4(CC), 4$
                      01 FB 00049    PUSHL #55
      05  FC  AB          02 E1 0004C 4$:    CALLS #1, FOR$$SIGNAL_STO
                      2F DD 00051    BBC #2, -4(CC), 5$
                      PUSHL #47

      1067
      1087
      1089
      1091
      1097
      1104
      1110
      1117
      1119
      1125
      1127

```

		63	01	FB 00053	CALLS #1, FOR\$\$SIGNAL_STO	
			5B	DD 00056	PUSHL CCB	1133
		64	01	FB 00058	CALLS #1, SY\$DELETE	
		3D	50	E8 0005B	BLBS R0, 11\$	
000182DA	8F	08	AB	D1 0005E	CMPL 8(CCB), #99034	1136
			10	12 00066	BNEQ 7\$	
			5B	DD 00068	PUSHL CCB	1138
00000000G	00		01	FB 0006A	CALLS #1, SY\$WAIT	
			5B	DD 00071	PUSHL CCB	1139
		64	01	FB 00073	CALLS #1, SY\$DELETE	
			E6	11 00076	BRB 6\$	1136
		1F	08	AB E8 00078	BLBS 8(CCB), 11\$	1141
000181A0	8F	08	AB	D0 0007C	MOVL 8(CCB), R0	1144
			50	D1 00080	CMPL R0, #98720	1146
000184B4	8F		09	13 00087	BEQL 8\$	
			50	D1 00089	CMPL R0, #99508	
			04	12 00090	BNEQ 9\$	
			35	DD 00092	PUSHL #53	
			02	11 00094	BRB 10\$	
			37	DD 00096	PUSHL #55	1147
		63	01	FB 00098	CALLS #1, FOR\$\$SIGNAL_STO	1144
00000000G			00	16 0009B	JSB FOR\$\$CB_POP	1155
			50	D4 000A1	CLRL R0	1157
			04	000A3	RET	1158
			0000	000A4	.WORD Save nothing	1067
		50	08	AC D0 000A6	MOVL 8(AP), R0	
		50	04	A0 D0 000AA	MOVL 4(R0), R0	
			F8	A0 9F 000AE	PUSHAB L_ERR_EQL_PRES	
			FC	A0 9F 000B1	PUSHAB L_UNWIND_ACTION	
			02	DD 000B4	PUSHL #2	
00000000G	00	7E	04	AC 7D 000B8	PUSHL SP	
			03	FB 000BC	MOVQ 4(AP), -(SP)	
			04	000C3	CALLS #3, FOR\$\$IOSTAT_HND	
					RET	

; Routine Size: 196 bytes, Routine Base: _FOR\$CODE + 0000

```
: 245      1159 1 GLOBAL ROUTINE FOR$DELETE_D (
: 246          1160 1     UNIT,                                ! Unit to delete on
: 247          1161 1     REC_NO,                               ! Record number to delete
: 248          1162 1     ERR_EQL,                             ! 1 if ERR= or IOSTAT= specified
: 249          1163 1     ) : CALL_CCB =
: 250
: 251          1165 1 ++ FUNCTIONAL DESCRIPTION:
: 252          1166 1 Deletes the specified record on a relative organization file
: 253          1167 1 opened for direct access.
: 254
: 255          1170 1 CALLING SEQUENCE:
: 256          1171 1
: 257          1172 1     iostat.wl.v = FOR$DELETE_D (unit.rl.v, rec_no.rl.v [, err_eql.r;u.v])
: 258
: 259          1173 1 FORMAL PARAMETERS:
: 260          1174 1
: 261          1175 1     unit
: 262          1176 1     rec_no
: 263          1177 1     err_eql
: 264          1178 1             - Unit number to delete on
: 265          1179 1             - Record number to delete
: 266          1180 1             - If present and 1, return IOSTAT
: 267          1181 1             values for errors.
: 268          1182 1 IMPLICIT INPUTS:
: 269          1183 1
: 270          1184 1     CCB
: 271          1185 1     LUBSL_LOG_RECNO
: 272          1186 1     LUBSV_FIND_LAST
: 273          1187 1             ! On if FIND was last operation
: 274          1188 1 IMPLICIT OUTPUTS:
: 275          1189 1
: 276          1190 1     LUBSV_FIND_LAST
: 277          1191 1             ! Cleared
: 278          1192 1 ROUTINE VALUE:
: 279          1193 1
: 280          1194 1     An IOSTAT value.
: 281          1195 1
: 282          1196 1 SIDE EFFECTS:
: 283          1197 1
: 284          1198 1     SIGNAL STOPs:
: 285          1199 1     FOR$_OPEDEFREQ - Open or define file required for direct or keyed access
: 286          1200 1     FOR$_WRIREAFIL - Write to read-only file
: 287          1201 1     FOR$_RECNUMOUT - Record number out of range
: 288          1202 1     FOR$_ATTACCNON - Attempt to access non-existent record
: 289          1203 1     FOR$_SPERECLOC - Specified record locked
: 290          1204 1     FOR$_DELEERR - Delete error
: 291          1205 1
: 292          1206 1     --
: 293          1207 1
: 294          1208 2     BEGIN
: 295          1209 2
: 296          1210 2     GLOBAL REGISTER
: 297          1211 2     CCB = 11: REF BLOCK [, BYTE];
: 298          1212 2
: 299          1213 2     LOCAL
: 300          1214 2     L_UNWIND_ACTION : VOLATILE,
: 301          1215 2     L_ERR_EQC_PRES : VOLATILE;
```

```
302      1216 2
303      1217 2
304      1218 2
305      1219 2
306      1220 2
307      1221 2
308      1222 2
309      1223 2
310      1224 2
311      1225 2
312      1226 2
313      1227 2
314      1228 2
315      1229 2
316      1230 2
317      1231 2
318      1232 2
319      1233 2
320      1234 2
321      1235 2
322      1236 2
323      1237 2
324      1238 2
325      1239 2
326      1240 2
327      1241 2
328      1242 2
329      1243 2
330      1244 2
331      1245 2
332      1246 2
333      1247 2
334      1248 2
335      1249 2
336      1250 2
337      1251 2
338      1252 2
339      1253 2
340      1254 2
341      1255 2
342      1256 2
343      1257 2
344      1258 2
345      1259 2
346      1260 2
347      1261 2
348      1262 2
349      1263 2
350      1264 2
351      1265 2
352      1266 2
353      1267 2
354      1268 2
355      1269 2
356      1270 2
357      1271 2
358      1272 2

        BUILTIN
          ACTUALCOUNT;

        ENABLE
          FOR$SIOSTAT_HND (L_UNWIND_ACTION, L_ERR_EQL_PRES);

        !+
        ! Determine if ERR_EQL is present.
        !-
        IF ACTUALCOUNT () GTR 2
        THEN
          L_ERR_EQL_PRES = .ERR_EQL
        ELSE
          L_ERR_EQL_PRES = 0;

        !+
        ! Unwind action is NO-OP (no LUB yet)
        !-
        L_UNWIND_ACTION= FORSK_UNWINDNOP;

        !+
        ! Get a LUB for this unit. On return, CCB points to the
        ! current control block.
        !-
        FOR$SCB_PUSH (.UNIT, LUBSK_LUN_MIN);

        !+
        ! Unwind action is now to POP the CCB.
        !-
        L_UNWIND_ACTION = FORSK_UNWINDPOP;

        !+
        ! If file is not relative organization, signal error FOR$_DELEERR.
        !-
        IF .CCB [LUB$B_ORGAN] NEQU LUBSK_ORG_RELAT
        THEN
          FOR$$SIGNAL_STO (FORSK_DELEERR);

        !+
        ! If file is read-only, signal error FOR$_WRIREAFIL.
        !-
        IF .CCB [LUB$V_READ_ONLY]
        THEN
          FOR$$SIGNAL_STO (FORSK_WRIREAFIL);

        !+
        ! If file is not direct access, signal error FOR$_OPEDEFREQ.
        !-
        IF NOT .CCB [LUB$V_DIRECT]
```

```
: 359      1273 2      THEN      FOR$SIGNAL_STO (FORSK_OPEDEFREQ);
: 360      1274 2
: 361      1275 2
: 362      1276 2
: 363      1277 2
: 364      1278 2
: 365      1279 2
: 366      1280 3      IF .REC_NO LEQ 0 OR (.CCB [LUBSL_REC_MAX] NEQ 0 AND .REC_NO GTRU .CCB [LUBSL_REC_MAX])
: 367      1281 2      THEN      FOR$SIGNAL_STO (FORSK_RECNUMOUT);
: 368      1282 2
: 369      1283 2
: 370      1284 2
: 371      1285 2      !+ See if the requested record to be deleted may already be locked.
: 372      1286 2      If it is, we want to avoid doing a $FIND which would unlock the
: 373      1287 2      record. To check, we compare REC_NO against (LUBSL_LOG_RECNO - 1)
: 374      1288 2      (or LUBSL_LOG_RECNO if LUBSV_FIND_LAST is set). If they are equal,
: 375      1289 2      the last operation we did may have locked this record. If it didn't,
: 376      1290 2      then we don't have a record locked. Trying to delete the record
: 377      1291 2      will show if it's locked or not. If not, we do a $FIND anyway.
: 378      1292 2
: 379      1293 2
: 380      1294 2
: 381      1295 3      IF .REC_NO EQL
: 382      ( IF .CCB [LUBSV_FIND_LAST]
: 383      THEN      .CCB [LUBSL_LOG_RECNO]
: 384      ELSE      .CCB [LUBSL_LOG_RECNO] - 1)
: 385      THEN      BEGIN
: 386      1300 2
: 387      1301 2
: 388      1302 2
: 389      1303 2      CCB [LUBSV_FIND_LAST] = 0;           ! Turn off bit
: 390      1304 2
: 391      1305 2
: 392      1306 2      !+ We may have it locked. Try to delete it.
: 393      1307 2
: 394      1308 3
: 395      1309 4      IF NOT $DELETE (RAB=.CCB)
: 396      THEN      WHILE .CCB [RABSL_STS] EQL RMSS_RSA DO
: 397      1311 3      BEGIN
: 398      1312 4      SWAIT (RAB=.CCB);
: 399      1313 4      $DELETE (RAB=.CCB);
: 400      1314 4      END;
: 401      1315 3
: 402      1316 3
: 403      1317 3      !+ If we succeeded, return.
: 404      1318 3
: 405      1319 3
: 406      1320 3
: 407      1321 3      IF .CCB [RABSL_STS]
: 408      THEN      BEGIN
: 409      1323 4      FOR$CB POP ();      ! Return I/O to previous state
: 410      1324 4      RETURN 0;          ! Success IOSTAT value
: 411      1325 4      END;
: 412      1326 3
: 413      1327 2
: 414      1328 2
: 415      1329 2      !+
```

```

416      1330 2    |_ We don't have it locked. Do a $FIND then a $DELETE.
417      1331 2
418      1332 2
419      1333 2
420      1334 2
421      1335 2
422      1336 2
423      1337 2
424      1338 3
425      1339 3
426      1340 4
427      1341 4
428      1342 5
429      1343 5
430      1344 5
431      1345 5
432      1346 5
433      1347 5
434      1348 5
435      1349 5
436      1350 5
437      1351 2
438      1352 2
439      1353 2
440      1354 2
441      1355 2
442      1356 2
443      1357 3
444      1358 2
445      1359 3
446      1360 3
447      1361 4
448      1362 4
449      1363 4
450      1364 3
451      1365 3
452      1366 3
453      1367 3
454      1368 2
455      1369 2
456      1370 2
457      1371 2
458      1372 2
459      1373 2
460      1374 2
461      1375 2
462      1376 2
463      1377 1

        CCB [LUB$V_FIND_LAST] = 0;
        CCB [LUB$L_LOG_RECNO] = .REC_NO;
        IF NOT $FIND (RAB=.CCB)
        THEN
          BEGIN
            WHILE .CCB[RAB$L_STS] EQL RMSS_RSA DO
              BEGIN
                SWAIT (RAB=.CCB);
                $FIND (RAB=.CCB)
              END;
              IF NOT .CCB [RAB$L_STS] THEN FOR$$SIGNAL_STO (
                SELECTONEU .CCB [RAB$L_STS] OF
                  SET
                    [RMSS_RLK] : FORSK_SPERELOC;
                    [RMSS_RNF, RMSS_EOF] : FORSK_ATTACNON;
                    [OTHERWISE] : FORSK_DELERR;
                  TES);
            END;

        |_ Try to delete the current record. If we get an error, signal it.

        IF NOT $DELETE (RAB=.CCB)
        THEN
          BEGIN
            WHILE .CCB [RAB$L_STS] EQL RMSS_RSA DO
              BEGIN
                SWAIT (RAB=.CCB);
                $DELETE (RAB=.CCB);
              END;
              IF NOT .CCB [RAB$L_STS]
              THEN
                FOR$$SIGNAL_STO (FORSK_DELERR);
            END;

        |_ Return I/O system to previous state
        |
        FOR$SCB_POP ();
        RETURN 0;           ! Success IOSTAT value
      END;

```

.EXTRN SYSS\$FIND

56 00000000G	00 087C 00000	.ENTRY FOR\$DELETE D, Save R2,R3,R4,R5,R6,R11
55 00000000G	00 9E 00002	MOVAB SYSS\$FIND, R6
54 00000000G	00 9E 00009	MOVAB SYSS\$WAIT, R5
		MOVAB SYSS\$DELETE, R4

: 1159
:

53	00000000G	00	9E	00017		MOVAB	FOR\$\$SIGNAL_STO, R3		
5E		04	C2	0001E		SUBL2	#4, SP	1208	
		AE	D4	00021		CLRL	L_ERR_EQL_PRES		
6D	0117	04	CF	DE	00023	CLRL	L_UNWIND_ACTION	1227	
02		6C	91	0002B		MOVAL	23\$, (FPT)		
		06	1B	0002E		CMPB	(APS), #2		
6E	OC	AC	D0	00030		BLEQU	1\$		
		02	11	00034		MOVL	ERR_EQL, L_ERR_EQL_PRES	1229	
04	AE	6E	D4	00036	1\$: 2\$:	BRB	2\$		
		01	D0	00038		CLRL	L_ERR_EQL_PRES	1231	
		50	D4	0003C		MOVL	#T, L_UNWIND_ACTION	1237	
52	04	AC	D0	0003E		CLRL	R0	1244	
	00000000G	00	16	00042		MOVL	UNIT, R2		
	04	AE	D4	00048		JSB	FOR\$\$CB_PUSH	1250	
02	C4	AB	91	0004B		CLRL	L_UNWIND_ACTION	1256	
		05	13	0004F		CMPB	-60(CCB), #2		
		37	DD	00051		BEQL	3\$		
FC	AB	01	FB	00053		PUSHL	#55	1258	
		02	E1	00056	3\$: 4\$:	CALLS	#1, FOR\$\$SIGNAL_STO		
		2F	DD	0005B		BBC	#2, -4(CCB), 4\$	1264	
FC	AB	01	FB	0005D		PUSHL	#4,	1266	
		04	E0	00060		CALLS	#1, FOR\$\$SIGNAL_STO	1272	
		1A	DD	00065		BBS	#4, -4(CCB), 5\$	1274	
63	52	01	FB	00067		PUSHL	#26		
	08	AC	D0	0006A	5\$: 6\$:	CALLS	#1, FOR\$\$SIGNAL_STO		
		0B	15	0006E		MOVL	REC_NO, R2	1280	
E4	AB	E4	AB	D5	00070	BLEQ	6\$		
		OB	13	00073		TSTL	-28(CCB)		
		OB	13	00073		BEQL	7\$		
E4	AB	52	D1	00075		CMPL	R2, -28(CCB)		
		05	1B	00079		BLEQU	7\$		
		19	DD	0007B	6\$: 7\$:	PUSHL	#25		
A0	63	01	FB	0007D		CALLS	#1, FOR\$\$SIGNAL_STO	1282	
	AB	03	E1	00080		BBC	#3, -96(CCB), 8\$	1295	
50		AB	D0	00085		MOVL	-32(CCB), R0	1297	
E0	AB	05	11	00089		BRB	9\$		
	50	01	C3	0008B	8\$: 9\$:	SUBL3	#1, -32(CCB), R0	1299	
	52	D1	00090			CMPL	R2, R0	1295	
A0	AB	26	12	00093		BNEQ	12\$		
	AB	08	8A	00095		BICB2	#8, -96(CCB)	1303	
		5B	DD	00099		PUSHL	CCB	1309	
64		01	FB	0009B		CALLS	#1, SYSSDELETE		
16		50	E8	0009E		BLBS	R0, 11\$		
000182DA	8F	08	AB	D1	000A1	10\$: 11\$:	CMPL	8(CCB), #99034	1311
		0C	12	000A9		BNEQ	11\$		
		5B	DD	000AB		PUSHL	CCB	1313	
65		01	FB	000AD		CALLS	#1, SYSSWAIT		
		5B	DD	000B0		PUSHL	CCB	1314	
64		01	FB	000B2		CALLS	#1, SYSSDELETE		
		EA	11	000B5		BRB	10\$		
A0	7D	08	AB	E8	000B7	11\$: 12\$:	BLBS	8(CCB), 22\$	1311
	AB	08	8A	000BB		BICB2	#8, -96(CCB)	1321	
EO	AB	52	D0	000BF		MOVL	R2, -32(CCB)	1333	
		5B	DD	000C3		PUSHL	CCB	1335	
66		01	FB	000C5		CALLS	#1, SYSSFIND	1336	
46		50	E8	000C8		BLBS	R0, 19\$		
000182DA	8F	08	AB	D1	000CB	13\$:	CMPL	8(CCB), #99034	1339

		OC	12	000D3	BNEQ	14\$		1341
		5B	DD	000D5	PUSHL	CCB		
		01	FB	000D7	CALLS	#1, SYSSWAIT		1342
		65			PUSHL	CCB		
		5B	DD	000DA	CALLS	#1, SYSSFIND		
		01	FB	000DC	BRB	13\$		
		66			BLBS	8(CC), 19\$		1344
		EA	11	000DF	MOVL	8(CC), R0		1345
		2C	AB	E8 000E1	14\$:	CMPL	R0, #98986	
		08			BNEQ	15\$		1347
000182AA	8F	50	D1	000E9	PUSHL	#52		
		04	12	000F0	BRB	18\$		
		34	DD	000F2	CMPL	R0, #98938		1348
		18	11	000F4	BEQL	16\$		
0001827A	8F	50	D1	000F6	15\$:	PUSHL	#98994	
		09	13	000FD	CMPL	R0, #98994		
000182B2	8F	50	D1	000FF	BNEQ	17\$		
		04	12	00106	PUSHL	#36		
		24	DD	00108	16\$:	BRB	18\$	
		02	11	0010A	CMPL	R0, #98938		
		37	DD	0010C	17\$:	PUSHL	#55	
		63	01	FB 0010E	18\$:	CALLS	#1, FOR\$\$SIGNAL_STO	
		5B	DD	00111	19\$:	PUSHL	CCB	
		64	01	FB 00113		CALLS	#1, SYSSDELETE	
		1F	50	E8 00116		BLBS	R0, 22\$	
000182DA	8F	08	AB	D1 00119	20\$:	CMPL	8(CC), #99034	
		0C	12	00121	BNEQ	21\$		1360
		5B	DD	00123	PUSHL	CCB		1362
		65	01	FB 00125		CALLS	#1, SYSSWAIT	
		5B	DD	00128	PUSHL	CCB		1363
		64	01	FB 0012A		CALLS	#1, SYSSDELETE	
		EA	11	0012D	BRB	20\$		1360
		05	AB	E8 0012F	21\$:	BLBS	8(CC), 22\$	
		37	DD	00133	PUSHL	#55		1365
		63	01	FB 00135		CALLS	#1, FOR\$\$SIGNAL_STO	
		00000000G	00	16 00138	22\$:	JSB	FOR\$\$CB_POP	
		50	D4	0013E	CLRL	R0		1374
		04	00140		RET			1377
			0000	00141	23\$:	.WORD	Save nothing	1208
		50	08	AC D0 00143	MOVL	8(AP), R0		
		50	04	A0 D0 00147	MOVL	4(R0), R0		
		F8	A0	9F 0014B	PUSHAB	L_ERR_EQL_PRES		
		FC	A0	9F 0014E	PUSHAB	L_UNWIND_ACTION		
		02	DD	00151	PUSHL	#2		
		5E	DD	00153	PUSHL	SP		
00000000G	00	7E	04	AC 7D 00155	MOVQ	4(AP), -(SP)		
		04	03	FB 00159	CALLS	#3, FOR\$\$IOSTAT_HND		
		04	00160		RET			

: Routine Size: 353 bytes, Routine Base: _FOR\$CODE + 00C4

FOR\$DELETE
1-002

K 16
16-Sep-1984 00:17:29 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 12:31:48 [FORRTL.SRC]FORDELETE.B32;1

Page 14
(5)

: 465 1378 1 END
: 466 1379 0 ELUDOM

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
_FOR\$CODE	549	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Symbols -----			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$_255\$DUA28:[SYSLIB]STARLET.L32;1	9776	13	0	581	00:01.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:FORDELETE/OBJ=OBJ\$:FORDELETE MSRC\$:FORDELETE/UPDATE=(ENH\$:FORDELETE)

: Size: 549 code + 0 data bytes
: Run Time: 00:16.4
: Elapsed Time: 00:55.0
: Lines/CPU Min: 5045
: Lexemes/CPU-Min: 33318
: Memory Used: 185 pages
: Compilation Complete

0179 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

COMRSOWD
LIS

FORDATEDS
LIS

FORDECOMO
LIS

FORCB
LIS

FORCLOSEF
LIS

FORDATE
LIS

COMSETST
LIS

FORASSO
LIS

FORDECOMF
LIS

FORDELETE
LIS

COMRADSO
LIS

COMUSEREX
LIS

FORBITOPS
LIS

FORDEFINE
LIS

FOROSPA

FORBACKSP
LIS

FOROUTR
LIS